

IN THE CLAIMS:

Please substitute the following claims for the same-numbered claims in the application:

1. (Currently Amended) A method of adaptively managing pages in a cache memory included within a system having a variable workload, said method comprising:
 - arranging a cache memory included within said system into a circular buffer;
 - maintaining a pointer that rotates around said circular buffer;
 - maintaining a skipping parameter value;
 - maintaining a bit for each page in said circular buffer, wherein a first bit value indicates that said page was not accessed by said system since a last time that said pointer traversed over said page, and a second bit value indicates that said page has been accessed since said last time said pointer traversed over said page; and
 - dynamically controlling a distribution of a number of pages in said cache memory that are marked with said first bit value in response to a variable workload in order to increase a hit ratio of said cache memory[[.]]; and
 - choosing pages to replace by moving said pointer around said circular buffer until a page that is marked with said first bit value is found, wherein while said pointer is moving around said circular buffer, if a page marked with said second bit value is found then remarking said page with said first bit value if since either a movement started or since a last page was marked, a determination is made that said skipping parameter value has been reached.

2. (Currently Amended) The method of claim 1, ~~further comprising:~~
~~maintaining a skipping parameter value; and~~
~~choosing pages to replace by moving said pointer around said circular buffer until a page~~
~~that is marked with said first bit value is found, wherein while said pointer is moving around said~~
~~circular buffer, if a page marked with said second bit value is found then remarking said page~~
~~with said first bit value if since either a movement started or since a last page was marked, a~~
~~determination is made that said skipping parameter value has been reached.~~
wherein the dynamic controlling process is self-controlling.

3. (Currently Amended) The method of claim 2 1, further comprising maintaining a FIFO history list, wherein said FIFO history list includes recently evicted pages from said circular buffer.

4. (Original) The method of claim 3, further comprising adaptively maintaining a target size for said FIFO history list that is exactly the number of pages that are marked with said second bit value at any given time.

5. (Original) The method of claim 4, further comprising maintaining a hit parameter value that is incremented when a hit is observed in said circular buffer on a page which is marked with said second bit value and decremented when a hit is observed on said FIFO history list.

6. (Original) The method of claim 5, further comprising adaptively determining said

skipping parameter value to ensure that the number of pages in said circular buffer that are marked with said second bit value are equal to said hit parameter value.

7. (Currently Amended) A system for adaptively managing pages in a cache memory included within said system having a variable workload comprising:

a circular buffer comprising said a cache memory;

a pointer that rotates around said circular buffer;

a skipping parameter value;

a bit marker operable for marking each page in said circular buffer with a bit value, wherein a first bit value indicates that said page was not accessed by said system since a last time that said pointer traversed over said page, and a second bit value indicates that said page has been accessed since said last time said pointer traversed over said page; and

a first controller operable for dynamically controlling a distribution of a number of pages in said cache memory that are marked with said first bit value in response to a variable workload in order to increase a hit ratio of said cache memory[.]; and

a second controller operable for choosing pages to replace by moving said pointer around said circular buffer until a page that is marked with said first bit value is found, wherein while said pointer is moving around said circular buffer, if a page marked with said second bit value is found then remarking said page with bit 0 if since either a movement started or since a last page was marked, a determination is made that said skipping parameter value has been reached.

8. (Currently Amended) The system of claim 7, further comprising:

a skipping parameter value; and

~~a second controller operable for choosing pages to replace by moving said pointer around said circular buffer until a page that is marked with said first bit value is found, wherein while said pointer is moving around said circular buffer, if a page marked with said second bit value is found then remarking said page with bit 0 if since either a movement started or since a last page was marked, a determination is made that said skipping parameter value has been reached.~~

wherein the dynamic controlling by said first controller is self-controlled.

9. (Currently Amended) The system of claim 8 7, further comprising a FIFO history list, wherein said FIFO history list includes recently evicted pages from said circular buffer.

10. (Original) The system of claim 9, further comprising a third controller operable for maintaining a target size for said FIFO history list that is exactly the number of pages that are marked with said second bit value at any given time.

11. (Original) The system of claim 10, further comprising a counter operable for maintaining a hit parameter value that is incremented when a hit is observed in said circular buffer on a page which is marked with said second bit value and decremented when a hit is observed on said FIFO history list.

12. (Original) The system of claim 11, further comprising a calculator operable for adaptively determining said skipping parameter value to ensure that the number of pages in said

circular buffer that are marked with said second bit value are equal to said hit parameter value.

13. (Currently Amended) A system for adaptively managing pages in a cache memory included within said system having a variable workload comprising:

means for arranging a cache memory included within a system into a circular buffer;

means for maintaining a pointer that rotates around said circular buffer;

means for maintaining a skipping parameter value;

means for maintaining a bit for each page in said circular buffer, wherein a first bit value indicates that said page was not accessed by said system since a last time that said pointer traversed over said page, and a second bit value indicates that said page has been accessed since said last time said pointer traversed over said page; and

means for dynamically controlling a distribution of a number of pages in said cache memory that are marked with bit 0 in response to a variable workload in order to increase a hit ratio of said cache memory[.]]; and

means for choosing pages to replace by moving said pointer around said circular buffer until a page that is marked with said first bit value is found, wherein while said pointer is moving around said circular buffer, if a page marked with said second bit value is found then remarking said page with said first bit value if since either a movement started or since a last page was marked, a determination is made that said skipping parameter value has been reached.

14. (Currently Amended) A program storage device readable by computer, tangibly embodying a program of instructions executable by said computer to perform a method of

adaptively managing pages in a cache memory included within a system having a variable workload, said method comprising:

arranging a cache memory included within a system into a circular buffer;

maintaining a pointer that rotates around said circular buffer;

maintaining a skipping parameter value;

maintaining a bit for each page in said circular buffer, wherein a first bit value indicates that said page was not accessed by said system since a last time that said pointer traversed over said page, and a second bit value indicates that said page has been accessed since said last time said pointer traversed over said page; and

dynamically controlling a distribution of a number of pages in said cache memory that are marked with said first bit value in response to a variable workload in order to increase a hit ratio of said cache memory[[.]]; and

choosing pages to replace by moving said pointer around said circular buffer until a page that is marked with said first bit value is found, wherein while said pointer is moving around said circular buffer, if a page marked with said second bit value is found then remarking said page with said first bit value if since either a movement started or since a last page was marked, a determination is made that said skipping parameter value has been reached.

15. (Currently Amended) The program storage device of claim 14, further comprising:

maintaining a skipping parameter value; and

choosing pages to replace by moving said pointer around said circular buffer until a page that is marked with said first bit value is found, wherein while said pointer is moving around said

~~circular buffer, if a page marked with said second bit value is found then remarking said page with said first bit value if since either a movement started or since a last page was marked, a determination is made that said skipping parameter value has been reached.~~

wherein the dynamic controlling process is self-controlling.

16. (Currently Amended) The program storage device of claim ~~15~~ 14, further comprising maintaining a FIFO history list, wherein said FIFO history list includes recently evicted pages from said circular buffer.

17. (Original) The program storage device of claim 16, further comprising adaptively maintaining a target size for said FIFO history list that is exactly the number of pages that are marked with said second bit value at any given time.

18. (Original) The program storage device of claim 17, further comprising maintaining a hit parameter value that is incremented when a hit is observed in said circular buffer on a page which is marked with said second bit value and decremented when a hit is observed on said FIFO history list.

19. (Original) The program storage device of claim 18, further comprising adaptively determining said skipping parameter value to ensure that the number of pages in said circular buffer that are marked with said second bit value are equal to said hit parameter value.

20. (Currently Amended) A method of adaptively managing pages in a cache memory included within a circular buffer in a system having a variable workload, said method comprising:

maintaining a pointer that rotates around said circular buffer;

maintaining a skipping parameter value;

assigning a bit value for the pages in the circular buffer, wherein a first bit value indicates that said pages have not been accessed by the system since a previous predetermined time, and wherein a second bit value indicates that said pages have been accessed by the system since a previous predetermined time;

converting said first bit value to said second bit value for selected pages; and

controlling a distribution of a number of pages in the cache memory that are marked with the first bit value in response to a variable workload in order to increase a hit ratio of the cache memory~~[[,]]~~; and

choosing pages to replace by moving said pointer around said circular buffer until a page that is marked with said first bit value is found, wherein while said pointer is moving around said circular buffer, if a page marked with said second bit value is found then remarking said page with said first bit value if since either a movement started or since a last page was marked, a determination is made that said skipping parameter value has been reached,

wherein the previous predetermined time includes the previous time said page was assigned said bit value.

21. (Currently Amended) The method of claim 20, ~~further comprising maintaining a pointer~~

~~that rotates around said circular buffer, wherein the controlling process is self-controlling.~~

22. (Currently Amended) The method of claim 21 20, wherein said first bit value indicates that a page was not accessed by said system since a last time that said pointer traversed over said page, and said second bit value indicates that said page has been accessed since said last time said pointer traversed over said page.

23. (Currently Amended) The method of claim 21, ~~further comprising:~~
~~maintaining a skipping parameter value; and~~
~~choosing pages to replace by moving said pointer around said circular buffer until a page~~
~~that is marked with said first bit value is found, wherein while said pointer is moving around said~~
~~circular buffer, if a page marked with said second bit value is found then remarking said page~~
~~with said first bit value if since either a movement started or since a last page was marked, a~~
~~determination is made that said skipping parameter value has been reached.~~
wherein said skipping parameter value is an adaptation parameter value.

24. (Original) The method of claim 23, further comprising maintaining a FIFO history list, wherein said FIFO history list includes recently evicted pages from said circular buffer.

25. (Original) The method of claim 24, further comprising adaptively maintaining a target size for said FIFO history list that is exactly the number of pages that are marked with said second bit value at any given time.

26. (Original) The method of claim 25, further comprising maintaining a hit parameter value that is incremented when a hit is observed in said circular buffer on a page which is marked with said second bit value and decremented when a hit is observed on said FIFO history list.

27. (Original) The method of claim 26, further comprising adaptively determining said skipping parameter value to ensure that the number of pages in said circular buffer that are marked with said second bit value are equal to said hit parameter value.

28. (Currently Amended) A system for adaptively managing pages in a cache memory included within a circular buffer in a system having a variable workload comprising:

means for maintaining a pointer that rotates around said circular buffer;

means for maintaining a skipping parameter value;

means for assigning a bit value for the pages in the circular buffer, wherein a first bit value indicates that said pages have not been accessed by the system since a previous predetermined time, and wherein a second bit value indicates that said pages have been accessed by the system since a previous predetermined time;

means for converting said first bit value to said second bit value for selected pages; and

means for controlling a distribution of a number of pages in the cache memory that are marked with the first bit value in response to a variable workload in order to increase a hit ratio of the cache memory[[],]; and

means for choosing pages to replace by moving said pointer around said circular buffer

until a page that is marked with said first bit value is found, wherein while said pointer is moving around said circular buffer, if a page marked with said second bit value is found then remarking said page with said first bit value if since either a movement started or since a last page was marked, a determination is made that said skipping parameter value has been reached,

wherein the previous predetermined time includes the previous time said page was assigned said bit value.